"Research on Development and Deployment of the Command and Control System Based on Service Components"

Topic 1: Concepts, Theory, and Policy

AUTHORS

QIU Hang-ping, LEI Zhi-peng, QUAN Ji-chuan

POC: QIU Hang-ping
PLA University of Science and Technology
Nanjing China, 13952004682
Qiuhp8887@hotmail.com

Authors' affiliations:
PLA University of Science and Technology

Report Documentation Page		*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **JUN 2014**	2. REPORT TYPE	3. DATES COVERED **00-00-2014 to 00-00-2014**

4. TITLE AND SUBTITLE **Research on Development and Deployment of the Command and Control System Based on Service Components**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **PLA University of Science and Technology,Nanjing China, 13952004682,**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 18th International Command & Control Research & Technology Symposium (ICCRTS) held 16-19 June, 2014 in Alexandria, VA. U.S. Government or Federal Rights License

14. ABSTRACT
In order to meet the demand for agility of future command and control (C2) system, the requirements of on-demand constructing, flexible reforming and adaptive adjusting should be achieved. In this paper, the whole-lifecycle of the C2 system is divided into three phases including component development, integration and deployment, operation and maintenance. The first phase is encapsulating the web services and legacy systems into service components through the component-based approaches. And during the second phase, on the basis of the capacity modeling analysis of the actual demand, the appropriate service components are selected. Thus, the C2 application systems will be built using the selected service components with the integrated system through a semi-automated manner. In the last phase, whenever specific requirements are proposed, the system can be adaptively adjusted or quickly upgraded by means of replacing or increasing service components dynamically. The service components are developed in distributed ways and published to public market to be used by all the system integrators. So we can avoid duplicate development and resources waste. Constructing systems through this method can shorten the development timeline effectively and improve the system flexibility and adaptability evidently.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

"Research on Development and Deployment of the Command and Control System Based on Service Components"

Topic 1: Concepts, Theory, and Policy

ABSTRACT

In order to meet the demand for agility of future command and control (C2) system, the requirements of on-demand constructing, flexible reforming and adaptive adjusting should be achieved. In this paper, the whole-lifecycle of the C2 system is divided into three phases including component development, integration and deployment, operation and maintenance. The first phase is encapsulating the web services and legacy systems into service components through the component-based approaches. And during the second phase, on the basis of the capacity modeling analysis of the actual demand, the appropriate service components are selected. Thus, the C2 application systems will be built using the selected service components with the integrated system through a semi-automated manner. In the last phase, whenever specific requirements are proposed, the system can be adaptively adjusted or quickly upgraded by means of replacing or increasing service components dynamically. The service components are developed in distributed ways and published to public market to be used by all the system integrators. So we can avoid duplicate development and resources waste. Constructing systems through this method can shorten the development timeline effectively and improve the system flexibility and adaptability evidently.

1 Introduction

With the advent of the information age, the role of C2 systems has become increasingly prominent. However, the long development cycle of the C2 systems is becoming the bottleneck of the development of the C2 systems. Meanwhile, with the continuous development of web technology, Service-oriented Architecture (SOA) has been widely used in C2 systems. The reusability and interoperability of the web services can enable the agile and rapid development of C2 systems.

With the use of SOA, Reference [1] examined some existing and emerging standards that can be applied to distribute vital information to remote users, and highlighted the benefits of using these standards to provide situational awareness and federated data displays. It has focused on two approaches: one is an architectural design pattern referred to as Representational State Transfer, and the other is based on what is commonly referred to as Web Services or WS* technology stack. In [2], the author showed that the main concerns of service-oriented C2 architecture is to improving interoperability of interdependent enterprises, business units, departments and agencies.

In this paper, through the introduction of the service component technology, the whole-lifecycle of the C2 system is divided into three phases including component development, integration and deployment, operation and maintenance. With daily development and accumulation of the service components and the service components-based system integration, it can shorten the C2 system development time

1

during the mission. So, the upgrades and updates of the C2 systems will become easier.

2 Service Components Technology

2.1 Definition of service component

Service component is a self-contained unit, which involves the management of service resources and encapsulates the corresponding functions and related data. Service component description and behavior are described by defined interface and attributes clearly. With the description, different service components can be combined to achieve new service components. The architecture of service component is shown in Figure 1. It consists of two parts: functional description documentation and functionality implementation file. Functional description documentation is written with XML format and brings about the functional description of Service components. It also provides the necessary configuration and modification of the service component attributes.

Functionality implementation file is a set of DLL files, which encapsulates the codes of the web service requests and responses, and implements the conversion from the web service data format to the local data format. It also exposes a set of interfaces to the local client to provide functionality.
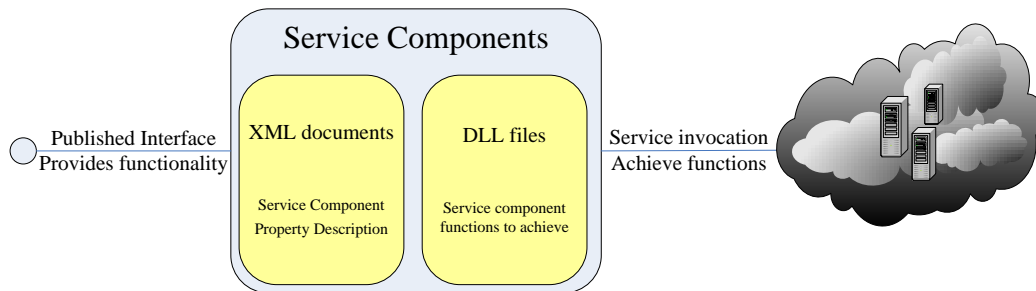

Figure 1 Structure Diagram of Service Component

2.2 Component-based Web service package

With the perspective of functions, web service can be considered as a distributed function. As the service invoking requires some certain specifications, the use of web services has high technical requirements. On the other hand, because the implementation of web services tends to be small granularity function, and in order to achieve the complete mission requirements, the composite service should be created by some appropriate web services. Therefore, by combining multiple granular forms of web services, the service components can achieve the functionality too. Meanwhile, encapsulation of service component also shields the internal details of web services, which reserves the space for the future change of the combination structure. By this way, the user can use the web service easier.

In detail, the encapsulation of service components is to package the complex invocation process of web service and translate the distributed function invocation into local process. The complex process of web service invocation is implemented by service components, so the users can use the distributed web services just like calling local functions. The invocation process of the service component is shown in Figure 2.
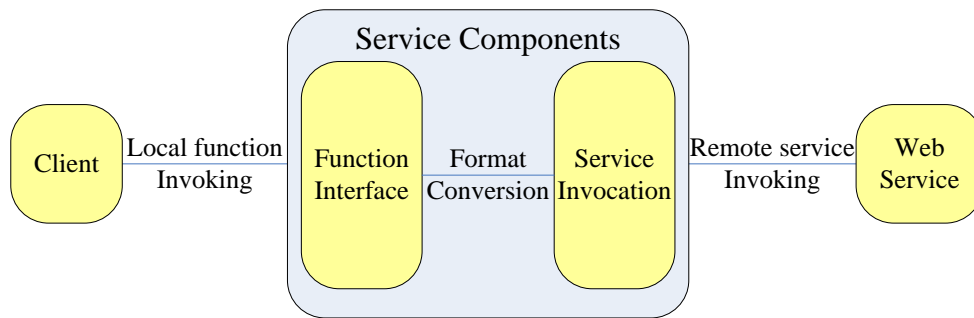
Figure 2 Schematic Diagram of the Service Package

In Figure 2, the main functional implementation of the service component consists of three parts: service invocation, function interface and format conversion.

Function interface is the entrance of the service component, which will be used by local users. The function and parameters specification of function interface are published by the description file and available to the users.

Service invocation module contains the web service invocation codes, which are created under the web service's rules.

The format conversion part is the bridge of the function interface and service invocation module. The major tasks of this part are to convert the major results of the web service into the local format and send it to the local function interface.

The main steps of the service component-based web service invocation are as follows.

1) The client calls the function interface of the local service component and passes parameters to the interface.

2) The function interface of the service component receives the parameters, and the format conversion part formats them with the web service type.

3) The service invocation module calls the web service by the parameters. The calling is in public format and asynchronous.

4) The service invocation module receives the results of the web service invoking, and the format conversion part changes the format of the data.

5) The formatted data are submitted to the client through the function interface.

2.3 Using the service components

When it is required to use service components to implement functional requirements, you can take the following steps.

First, according to the requirements of the mission, the parameters are modified to meet the user's requirements.

Secondly, the service components are loaded dynamically. In the phase of system integration, the main work is to compose multiple service components into the complete C2 system by the system integration platform.

3 Integration and deployment of C2 system

With the development of information technology, all kinds of information systems have appeared, and they play important roles in diverse systems. However, the endless lifecycle of the traditional development approach of information systems can't meet the users' requirements. For example, the C2 system for earthquake must meet the following

3

requirements:

1) Rapid Building. Whenever we get the task, the system must be constructed quickly.

2) Easy Building. The system users should be enabled to build the system quickly after a simple training.

3) Convenient Adjusting. Whenever the environment changes, the system can be adjusted by users to meet the changes.

4) Flexible Deployment. The system can be easily deployed in cross-platforms, such as desktops, notebooks, PDAs and other types of terminals.

3.1 Service components based C2 system

The structure of the service components-based C2 system is divided into four layers: services layer, components layer, platform layer and application layer, as shown in Figure 3.
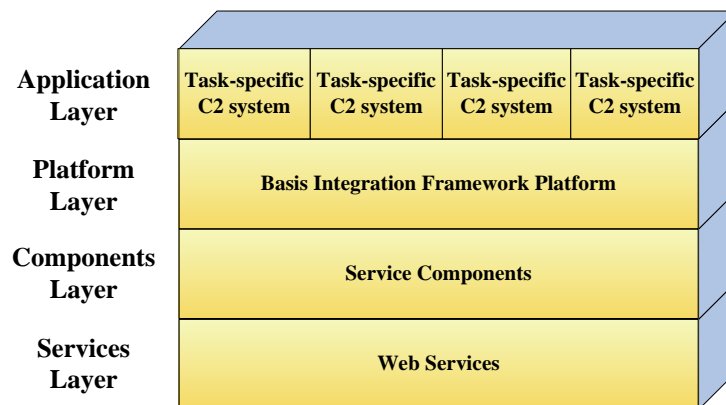

Figure 3 the Architecture of Service Components-Based C2 System

1) Services Layer: This layer contains all web services which are known and published on the web. The legacy systems which have been packaged into services are also included.

2) Components Layer: This layer includes diverse service components, which are either compositions of some web services, or combinations of some small granularity service components. The common characteristic of the service components is the ability to be invoked by the basis integration framework platform to implement the basic functions.

3) Platform Layer: The basis integration framework platform is the execution basis of the service components-based system. The platform itself does not provide any functionality. All the capacities for C2 are implemented by the service components. The platform only provides the execution environment for the service components.

4) Application Layer: This layer contains the exclusive C2 systems of specific tasks. It can be written by professionals directly as traditional way does. Also it can be quickly integrated by the system user through the integrated configuration tools.

3.2 Basis integration framework Platform

As the core part of the system, basis integration framework platform itself does not provide any C2 related functionality, it only implements the main and basic modules of the system, such as the cluster information sharing modules, the dynamic loading modules for service components and other basic functional modules.

1) Cluster Information Sharing Modules

For the information sharing module, there are many optional network structures. Since the

responses of the web services operations and data acquisition by the client are very similar, we choose the bus structure as the basic structure. As shown in figure 4, all the information should be published into the bus directly, and be pushed to all terminals. Each terminal may receive and process the information with its own requirements.

2) Service Components Dynamic Loading Modules

During the system runtime, whenever a new capacity is needed, the basis platform can load the implementation DLL files dynamically to meet the new requirements. The DLLs are described by service components description documentation and also used with it.
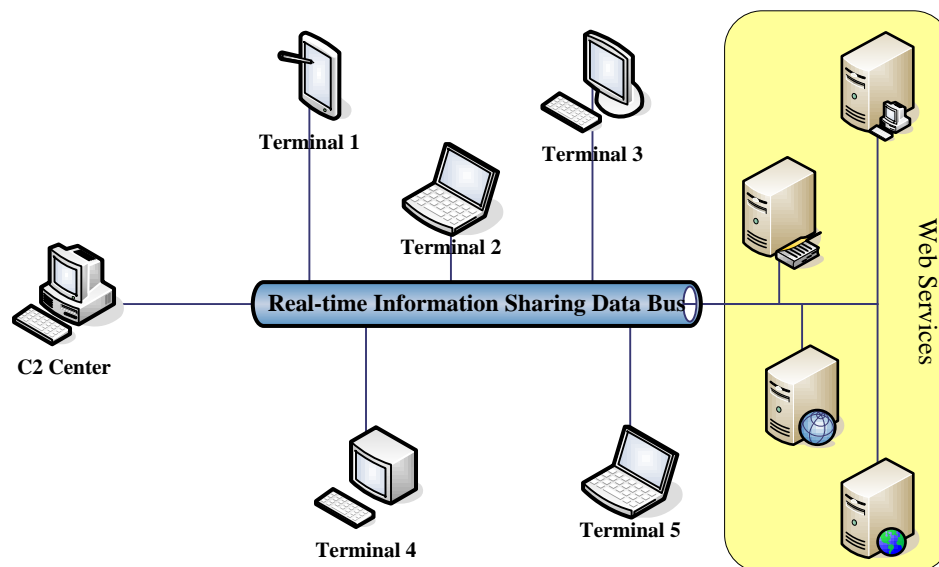

Figure 4 Real-time information sharing module

3.3 Building steps for command and control system

The service component-based C2 system is designed and implemented as follows:

Step 1: Prepare the service components and verify the validity of web services used by service components. The approaches to obtain services mainly have three ways. First, it can be selected from the local component library according to the demands. Secondly, it can be created by combining the components from the local component library. Finally, development may be necessary for the specific needs.

Step 2: Each group exchanges its local component library with other group and uses the same basis integration framework platform.

Step 3: The main group, usually the command group, runs the system components integration and configuration tool and generates the task-specific system components basing on mission requirements.

The main work in system integration phase is to choose the appropriate service components and integrate them into a large-grained service component. In this phase, the real C2 system has not come out. The complete one will not appear until the system is published.

3.4 System Features

The integration of the service components-based C2 system has the following characteristics:

1) Establishment and richness of the local component library have a direct impact on steps 1.

Over time, when the local component library has accumulated enough components, the system integration of each task may only need to find the local component library for its components. By this way, the system integration time will be greatly decreased and the efficiency of the system construction will be undoubtedly greatly increased.

2) Since the system constructed by this way is built based on the task at run time, it will result in a good adaptability and meet the demands.

3) Through the development and use of the data fusion components and real-time information sharing components, the system can achieve standardized package of different data sources and integration of them. Thus the service components-based system has a good real-time information sharing capabilities.

4 Operation and maintenance of C2 system

4.1 System operation

When the task group arrives at the target area, the group will run the basis integration framework platform and connect to the network to find other clients. The first client is responsible for the loading of the system components, and the other clients will get the system components from the network when they start up. After the system is online and the system components are loaded, the clients can modify the system according to their own conditions and requirements.

With the transformation of the battlefield situation, when a client needs to add new capacity to the system, it may only select the appropriate service components, amend them and add to the system integrated framework dynamically. The changes can be synchronized through the network in real-time.

4.2 System Maintenance

As the web services are executed in a dynamic network environment, they have greater instability. In the runtime, the instability of distributed web services may cause unexpected failure and lead to failure of the entire system, which becomes a significant risk of web services-based systems. But the ability that service components can be dynamically replaced in this framework will achieve a good recovery capability from the system failures.

Meanwhile, this platform has the ability to add or replace components dynamically, which will greatly enhance the system's adaptability and survivability in battlefield.

5 Conclusions

By introducing the service component technology, this paper presents a service component-based C2 system integration and deployment method. Through analysis and comparison, this method has the following four features:

1) System construction is convenient, fast and efficient.
2) Good data service integration and real-time information sharing capabilities.
3) Good environment adaptability and battlefield survivability.
4) The system has the ability to update and upgrade dynamically.

6 Reference

[1] Gette, Dennis L. Emerging service oriented architecture-based concepts in air traffic control[C]. 51st Air Traffic Control Association Annual Conference and Exposition 2006, 2006.

[2] Bayne, J.S. A Theory of Enterprise Command and Control[C]. Military Communications Conference 2006, 2006.